

Time limit	1 s
Memory limit	64 MB

----HEWAN

Terdapat sebuah base class bernama **Hewan**. Base class tersebut memiliki dua derived class yaitu **Kucing** dan **Anjing**. Selain itu terdapat juga derived class dari class **Kucing** yaitu **KucingAnggora**, dan terdapat juga derived class dari class **Anjing** yaitu **AnjingBulldog**. Setiap class hanya memiliki satu **constructor** dengan parameter dan satu **destructor**.

Untuk class Hewan:

1. Memiliki satu attribute bertipe string yang bersifat protected yaitu **nama**.
2. Memiliki constructor dengan **satu parameter** yaitu string yang merupakan nama dari hewan. Parameter constructor **tidak** menggunakan reference. Constructor **tidak** menghasilkan keluaran apapun.
3. Memiliki dtor yang akan **memberikan keluaran** tentang kematian objek seperti pada contoh kasus dibawah.

Untuk class Anjing dan Kucing:

1. Memiliki constructor dengan **satu parameter** yaitu string yang merupakan nama dari hewan. Parameter constructor **tidak** menggunakan reference. Constructor **tidak** menghasilkan keluaran apapun.
2. Memiliki dtor yang akan **memberikan keluaran** tentang kematian objek seperti pada contoh kasus dibawah.

Untuk class KucingAnggora dan AnjingBulldog:

1. Memiliki satu attribute bertipe string yang bersifat protected yaitu **pemilik**.
2. Memiliki constructor dengan **dua parameter** yaitu string yang merupakan nama dari hewan dan string yang merupakan nama dari pemilik hewan. Parameter constructor **tidak** menggunakan reference. Constructor **tidak** menghasilkan keluaran apapun.
3. Memiliki dtor yang akan **memberikan keluaran** tentang kematian objek seperti pada contoh kasus dibawah.

Perlu diperhatikan objek apapun yang dihasilkan dari base class Hewan dan derived classnya dipastikan dapat memanggil fungsi **void print()** yang akan menuliskan keluaran sesuai dengan studi kasus dibawah. Kompetensi yang diuji pada soal ini ialah pemahaman mahasiswa terhadap materi **Polymorphism**. Melalui soal ini diharapkan mahasiswa dapat memahami penggunaan fungsi **virtual** serta **dynamic binding**.

Silahkan membuat definisi dan implementasi dari class **Hewan**, **Kucing**, **Anjing**, **KucingAnggora**, dan **AnjingBulldog** sehingga **main program** berikut

```
int main() {
    Hewan *array_hewan[3];
    Hewan a("a");
    Kucing b("b");
    Anjing c("c");
    array_hewan[0] = &a;
    array_hewan[1] = &b;
    array_hewan[2] = &c;

    for(int i=0;i<3;i++) {
        array_hewan[i]->print(); //Keluaran = "[Makhluk] ini namanya [nama]\n"
    }
}
```

```
array_hewan[1] = &c;
array_hewan[1]->print();
```

```
Kucing *d = new KucingAnggora("d","1");
Anjing *e = new AnjingBulldog("e","2");
d->print();
e->print();
KucingAnggora f("f","3");
AnjingBulldog g("g","4");
f.print();
g.print();
delete e;
delete d;
return 0;
}
```

Dapat menghasilkan **keluaran seperti dibawah**, abaikan tulisan dalam {}, tulisan tersebut merupakan **hint**.


```

Hewan ini namanya a{dari fungsi print}
Hewan ini namanya b{dari fungsi print}
Anjing ini namanya c{dari fungsi print}
Anjing ini namanya c{dari fungsi print}
Kucing anggora ini namanya d. Pemiliknya adalah 1{dari fungsi print}
Anjing bulldog ini namanya e. Pemiliknya adalah 2{dari fungsi print}
Kucing anggora ini namanya f. Pemiliknya adalah 3{dari fungsi print}
Anjing bulldog ini namanya g. Pemiliknya adalah 4{dari fungsi print}
Anjing mati e{dari destructor}
Hewan mati e{dari destructor}
Kucing anggora mati d{dari destructor}
Kucing mati d{dari destructor}
Hewan mati d{dari destructor}
Anjing bulldog mati g{dari destructor}
Anjing mati g{dari destructor}
Hewan mati g{dari destructor}
Kucing anggora mati f{dari destructor}
Kucing mati f{dari destructor}
Hewan mati f{dari destructor}
Anjing mati c{dari destructor}
Hewan mati c{dari destructor}
Kucing mati b{dari destructor}
Hewan mati b{dari destructor}
Hewan mati a{dari destructor}

```

Kumpulkan **Hewan.h** yang merupakan deklarasi sekaligus definisi dari semua class (Hewan, Kucing, Anjing, KucingAnggora, AnjingBulldog) dalam satu file. Berikut merupakan contoh satu file [contoh.h](#) yang berisi multiclass.

C++14 ↕

 [Hewan.h](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	100	Accepted	0.00 sec, 2.87 MB

Time limit	1 s
Memory limit	64 MB

Menurut Wikipedia, **Rekening bank** adalah rekening keuangan pencatatan transaksi keuangan antara pelanggan dan bank mereka dan posisi keuangan yang dihasilkan dari pelanggan dengan bank.

Rekening bank pada umumnya dibagi menjadi dua jenis, yaitu Rekening Giro dan Rekening Tabungan. **Rekening Giro** adalah rekening bank yang menatausahakan dana yang sewaktu-waktu dapat ditarik atau disetor oleh nasabah Sedangkan **Rekening Tabungan** adalah rekening yang dikelola oleh nasabah pada bank untuk tujuan mengumpulkan dana

Diberikan sebuah *header* kelas **Rekening**, **RekeningGiro**, **RekeningTabungan** sebagai berikut.

```
// Rekening.h
#ifndef REKENING_H
#define REKENING_H

class Rekening {
public:
    // Konstruktor untuk menginisialisasi saldo
    // Set saldo 0.0 apabila saldo bernilai negatif
    Rekening(double);

    // Getter, Setter
    void setSaldo(double);
    double getSaldo() const;

    // Member Function
    // Tambahkan sejumlah nominal uang ke saldo
    void simpanUang(double);

    // Kurangkan sejumlah nominal uang dari saldo apabila saldo mencukupi (tidak negatif setelah dikurangkan)
    // Kembalikan boolean yang mengindikasikan apakah uang berhasil ditarik atau tidak
    bool tarikUang(double);

private:
    double saldo;
};

#endif
```

```
// RekeningGiro.h
#ifndef REKENINGGIRO_H
#define REKENINGGIRO_H

#include "Rekening.h" // include the Account header file

class RekeningGiro : public Rekening {
public:
    // Konstruktor menginisialisasi saldo (parameter 1) dan suku bunga (parameter 2)
    // Set suku bunga 0.0 apabila suku bunga bernilai negatif
    RekeningGiro(double, double);

    // Getter, Setter
    void setSukuBunga(double);
    double getSukuBunga() const;

    // Member Function
    // Bunga dihitung dari saldo dikali suku bunga
    double hitungBunga();

private:
    double sukuBunga;
};

#endif
```

```
// RekeningTabungan.h
#ifndef REKENINGTABUNGAN_H
#define REKENINGTABUNGAN_H

#include "Rekening.h"

class RekeningTabungan : public Rekening {
public:
    // Konstruktor menginisialisasi saldo (parameter 1) dan biaya transaksi (parameter 2)
    // Set biaya transaksi 0.0 apabila biaya transaksi bernilai negatif
    RekeningTabungan(double, double);

    // Getter, Setter
    void setBiayaTransaksi(double);
    double getBiayaTransaksi() const;

    // Member Function
    // Panggil fungsi simpanUang dari parent class
    // Kurangkan saldo dengan biaya transaksi
    void simpanUang(double);

    // Panggil fungsi tarikUang dari parent class
    // Kurangkan saldo dengan biaya transaksi hanya jika penarikan uang berhasil
    // Saldo mungkin saja menjadi negatif apabila setelah penarikan, saldo < biaya transaksi
    // Kembalikan boolean yang mengindikasikan apakah uang berhasil ditarik atau tidak
    bool tarikUang(double);

private:
    double biayaTransaksi;
};

#endif
```

Kumpulkan file **Rekening.cpp**, **RekeningGiro.cpp**, **RekeningTabungan.cpp** yang dimasukkan ke dalam zip bernama **Rekening_<nim>.zip**.

C++14 ↕

 [Rekening_13522091.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	7	Accepted	0.00 sec, 3.05 MB
2	7	Accepted	0.00 sec, 3.13 MB
3	7	Accepted	0.00 sec, 3.03 MB
4	7	Accepted	0.00 sec, 3.13 MB
5	8	Accepted	0.00 sec, 3.05 MB
6	8	Accepted	0.00 sec, 3.06 MB
7	8	Accepted	0.00 sec, 2.93 MB
8	8	Accepted	0.00 sec, 3.02 MB
9	8	Accepted	0.00 sec, 2.93 MB
10	8	Accepted	0.00 sec, 3.07 MB

No	Score	Verdict	Description
11	8	Accepted	0.00 sec, 3.13 MB
12	8	Accepted	0.00 sec, 3.05 MB
13	8	Accepted	0.00 sec, 3.05 MB

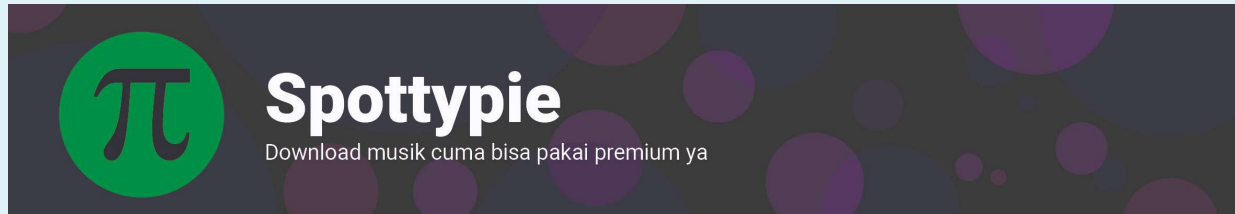
Question **3**

Correct

Mark 90.00 out of 90.00

Time limit	1 s
Memory limit	64 MB

Baru saja beberapa hari setelah hari Valentine berakhir. Masih banyak anak muda mudi pergi bersama untuk menikmati kebersamaan karena sibuk mengerjakan tugas besar, diperlukan rekreasi yang bisa dilakukan bersamaan dengan mengerjakan tugas besar. Oleh karena itu, Spottypie solusinya.



Spottypie adalah aplikasi yang dapat digunakan untuk memainkan musik secara *online*. Pengguna aplikasi Spottypie terdiri dari tiga jenis, yaitu pengguna biasa, pengguna premium, dan artis/musisi. Pengguna biasa hanya dapat mendengarkan musik dan menyimpannya dalam daftar musik favorit. Pengguna premium dapat mengunduh musik sehingga dapat dijalankan secara *offline*. Pengguna artis dapat mengunggah musik.

Anda diminta untuk membuat "aplikasi" kecil sejenis Spottypie. Yang perlu anda buat adalah implementasi kelas dalam **.cpp** saja. Header dari masing-masing kelas dapat diunduh melalui tautan berikut ini.

- [User.h](#)
- [PremiumUser.h](#)
- [ArtistUser.h](#)

Catatan:

Gunakan library **<cstring>** untuk menggunakan fungsi-fungsi pengolah string:

- **strcmp(str1, str2)** untuk membandingkan str1 dan str2, jika hasil 0 berarti kedua string sama.
- **strcpy(str1, str2)** untuk menyalin str2 ke str1
- **strlen(str)** untuk memeriksa panjang sebuah string

Upload sebuah **zip** berisi file-file **User.cpp**, **PremiumUser.cpp**, dan **ArtistUser.cpp**.

C++14

 [Spottypie.zip](#)

Score: 90

Blackbox

Score: 90

Verdict: Accepted

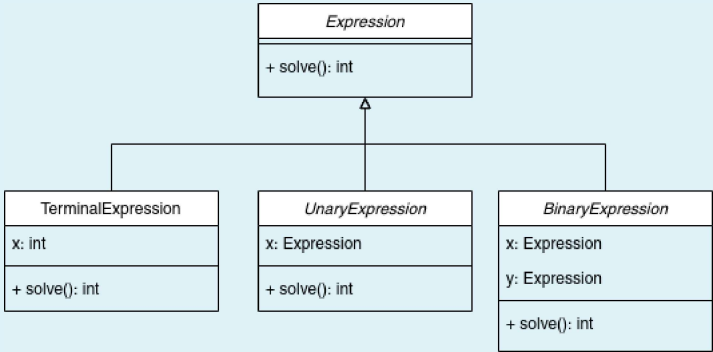
Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.88 MB
2	10	Accepted	0.00 sec, 2.87 MB
3	10	Accepted	0.00 sec, 2.94 MB
4	10	Accepted	0.01 sec, 2.89 MB

No	Score	Verdict	Description
5	10	Accepted	0.00 sec, 3.00 MB
6	10	Accepted	0.00 sec, 2.89 MB
7	10	Accepted	0.00 sec, 3.00 MB
8	10	Accepted	0.00 sec, 2.87 MB
9	10	Accepted	0.00 sec, 2.89 MB

Time limit	1 s
Memory limit	64 MB

Diberikan empat buah class Expression, UnaryExpression, BinaryExpression, dan TerminalExpression. UnaryExpression merupakan ekspresi matematika yang menerima 1 parameter, BinaryExpression menerima 2 parameter, dan TerminalExpression merepresentasikan sebuah angka. Berikut file [Expression.hpp](#) dan [Expression.cpp](#) (untuk memudahkan, semua definisi dan implementasi digabung ke file yang sama, tapi jangan ditiru ya!).



Anda diminta untuk membuat class AddExpression dan SubtractExpression yang diturunkan dari BinaryExpression, serta class NegativeExpression yang diturunkan dari UnaryExpression.

Contoh Main	Contoh Output
<pre>e = new AddExpression(new TerminalExpression(4), new TerminalExpression(6)); e->solve()</pre>	10
<pre>e = new SubtractExpression(new TerminalExpression(4), new TerminalExpression(6)); e->solve()</pre>	-2
<pre>e = new NegativeExpression(new TerminalExpression(1)); e->solve()</pre>	-1

Kumpulkan dalam **Expression.zip** berisi AddExpression.hpp, AddExpression.cpp, SubtractExpression.hpp, SubtractExpression.cpp, NegativeExpression.hpp, dan NegativeExpressio.cpp!

C++14

[Expression.zip](#)

Score: 50

Blackbox

Score: 50

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.87 MB

No	Score	Verdict	Description
2	10	Accepted	0.00 sec, 2.91 MB
3	10	Accepted	0.00 sec, 2.85 MB
4	10	Accepted	0.00 sec, 2.97 MB
5	10	Accepted	0.00 sec, 2.85 MB

[◀ Slide Tutorial 2](#)

Jump to...

